

Title: Creating an Application in Salesforce.com using Apex programming Language

CreateAccount.apxc

```
public class CreateAccount {
    public String name {get; set;}
    public String phoneNumber {get; set;}
    public String selectedname {get; set;}
    public String websiteURL {get; set;}
    public List<Selectoption> lstnamesel {get; set;}

    public CreateAccount(ApexPages.StandardController controller) {
        lstnamesel = new List<selectoption>(); lstnamesel.add(new
        selectOption('-', 'None -')); lstnamesel.add(new
        selectOption('IT', 'IT')); lstnamesel.add(new
        selectOption('MECH', 'MECH')); lstnamesel.add(new
        selectOption('CHEM', 'CHEM')); lstnamesel.add(new
        selectOption('PHARMA', 'PHARMA'));
    }

    public PageReference createAccount() {
        System.debug('teset create');
        if(!String.isEmpty(name)) {
            Account accountRecord = new Account(Name = name,
                Phone = phoneNumber,
                Industry = selectedname,
                Website = websiteURL);

            INSERT accountRecord;

            PageReference pg = new PageReference('/'+accountRecord.Id);
            pg.setRedirect(true);
            return pg;
        } else {
            ApexPages.addmessage(new ApexPages.message(ApexPages.severity.CONFIRM, 'Please enter
            Account Name'));
        }
        return NULL;
    }

    public PageReference cancelAccount() {
        PageReference pg = new PageReference('/'+Schema.SObjectType.Account.getKeyPrefix()+'/o');
```

```

        return pg;
    }
}

```

CreateAccount.vfp

```

<apex:page standardController="Account"
    extensions="CreateAccount"> <apex:form id="apexForm" >
    <apex:pageBlock title="Create Account:" id="pageBlockId">

        <apex:pageMessages id="showmsg"></apex:pageMessages>
        <apex:pageBlockSection columns="2" >
            <!--<div class = "requiredInput">
                <div class = "requiredBlock"></div>
                <apex:inputText value="{!name}" label="Account Name"
                    required="true"/> </div>-->
            <apex:inputText value="{!name}" label="Account Name" required="true"/>
            <apex:inputText value="{!phoneNumber}" label="Phone" /> <apex:selectList
                size="1" value="{!selectedname}" label="Industry">
                <apex:selectOptions value="{!lstnamesel}"/>
            </apex:selectList>
            <apex:inputText value="{!websiteURL}" label="Website"
        /> </apex:pageBlockSection>

        <apex:pageBlockButtons >
            <apex:commandButton action="{!createAccount}" value="Save"/>
            <apex:commandButton action="{!cancelAccount}" value="Cancel"/>
        </apex:pageBlockButtons>
    </apex:pageBlock>

    <!-- ACTION FUNCTIONS-->
    <apex:actionFunction name="createFunction" action="{!createAccount}"/>
    <apex:actionFunction name="cancelFunction" action="{!cancelAccount}"/>
</apex:form>
</apex:page>

```

Step No 1:

Create new org:

<https://developer.salesforce.com/signup>

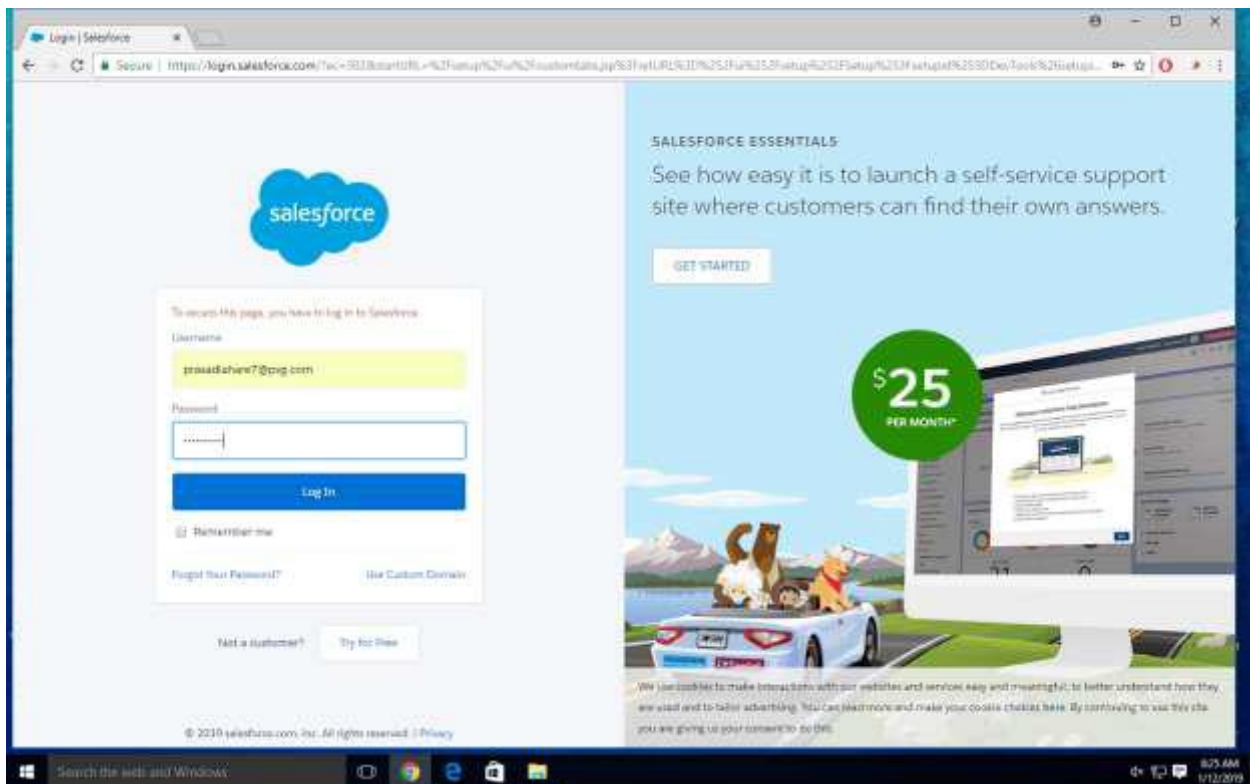
Step No 2:

After signup, logging using following URL

<https://login.salesforce.com/>

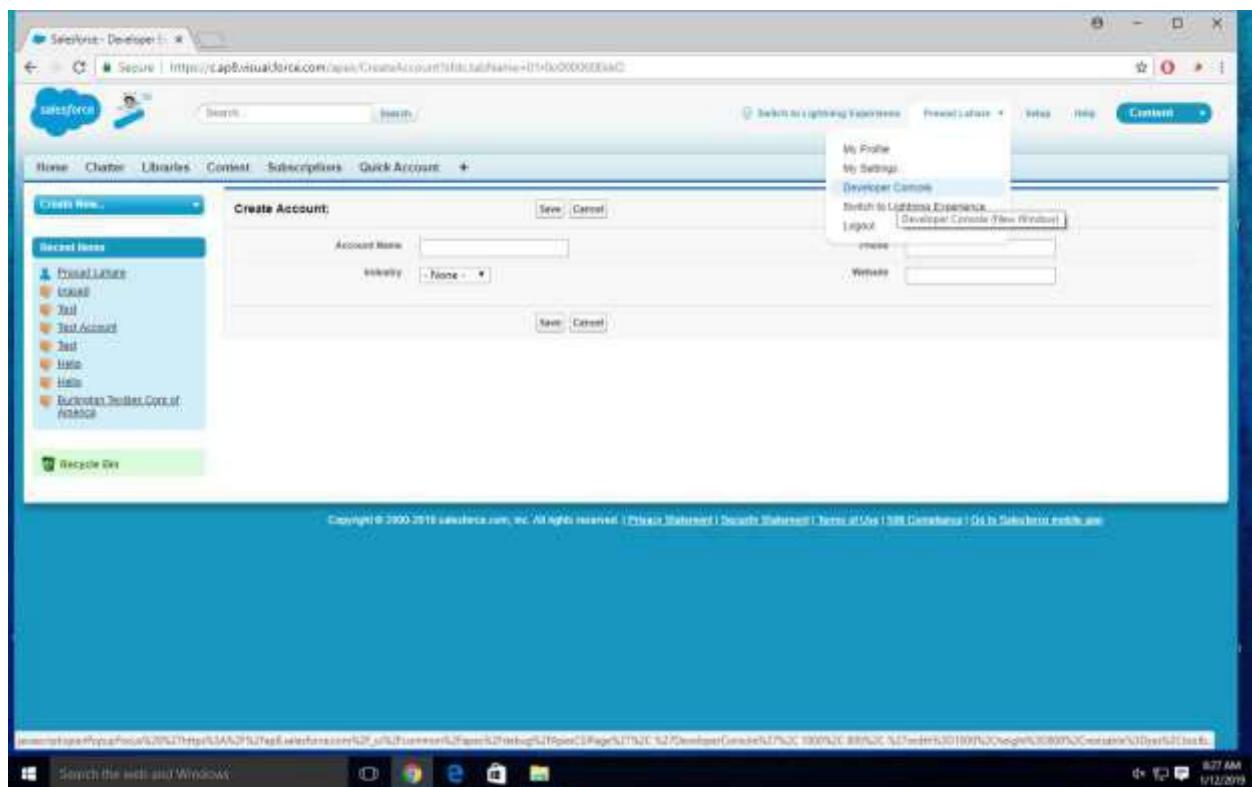
Step No 3:

Login Page



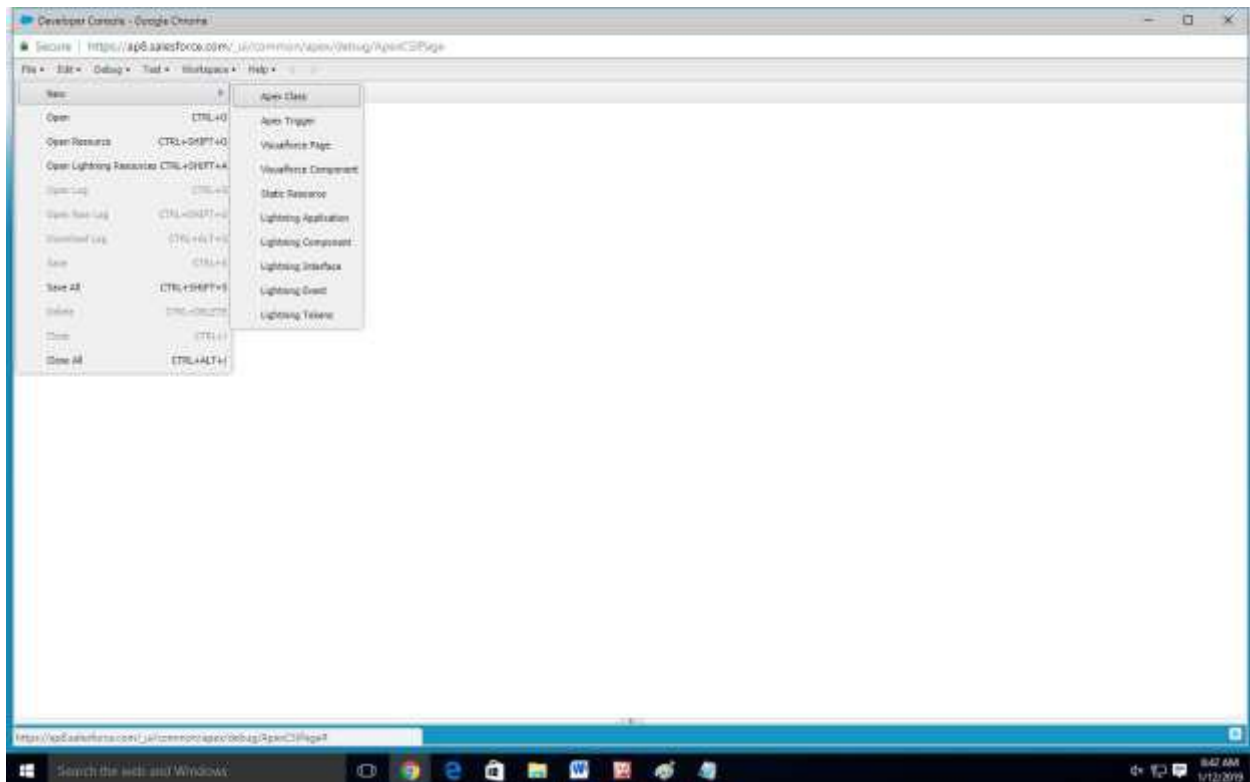
Step No:4

Go to Developer Console for writing Program



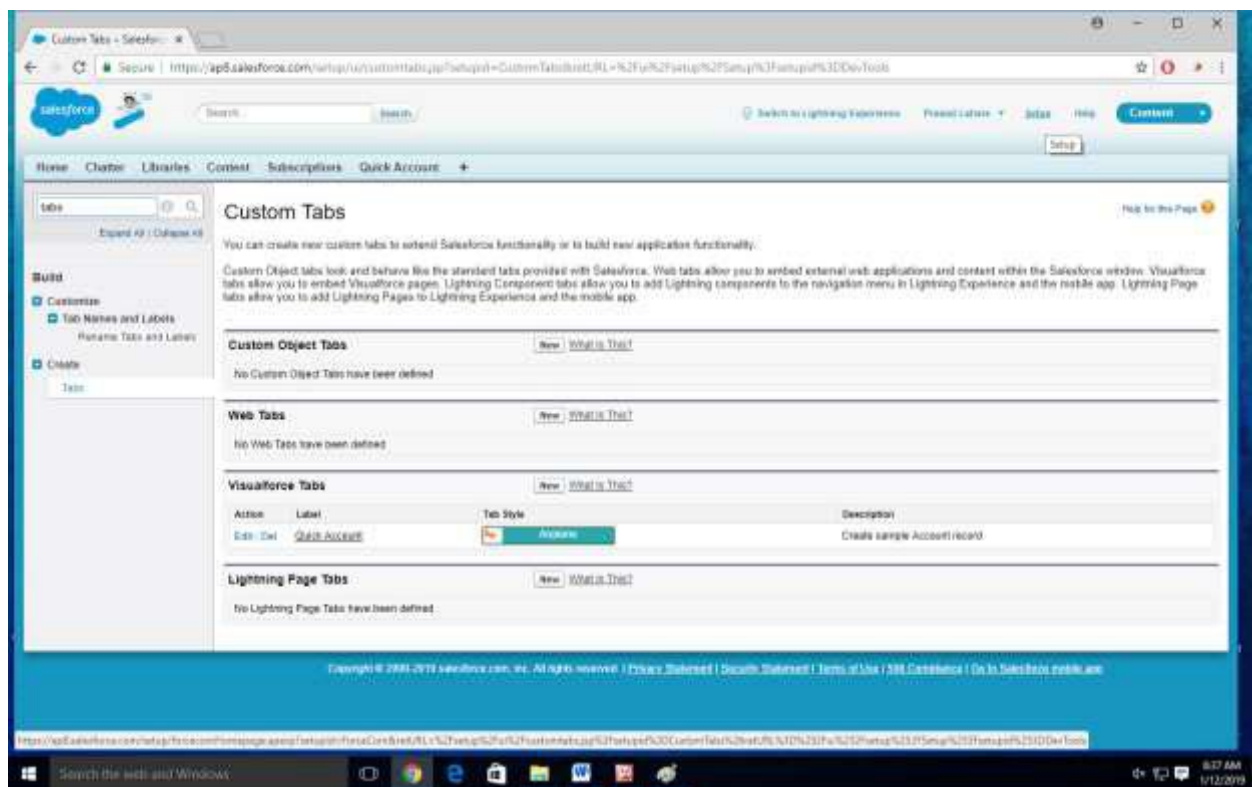
Step No 5:

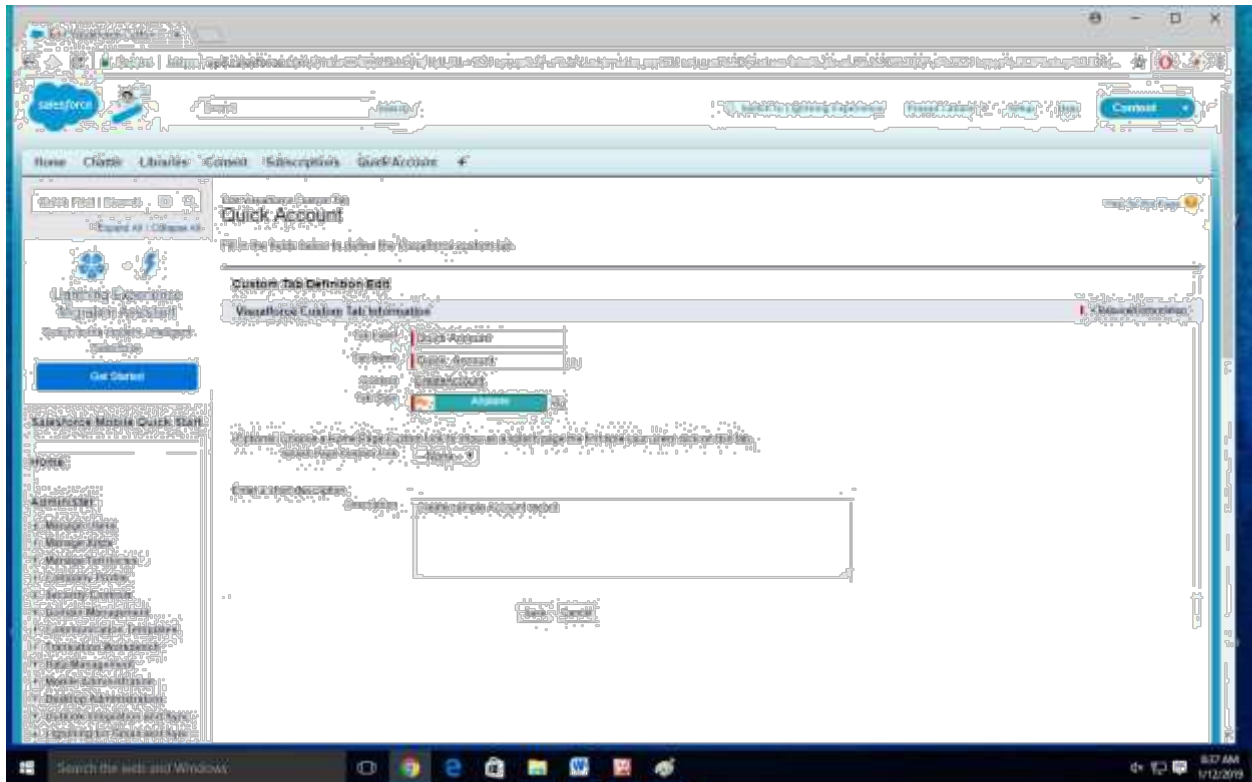
Click on file then Select Apex class and write a code



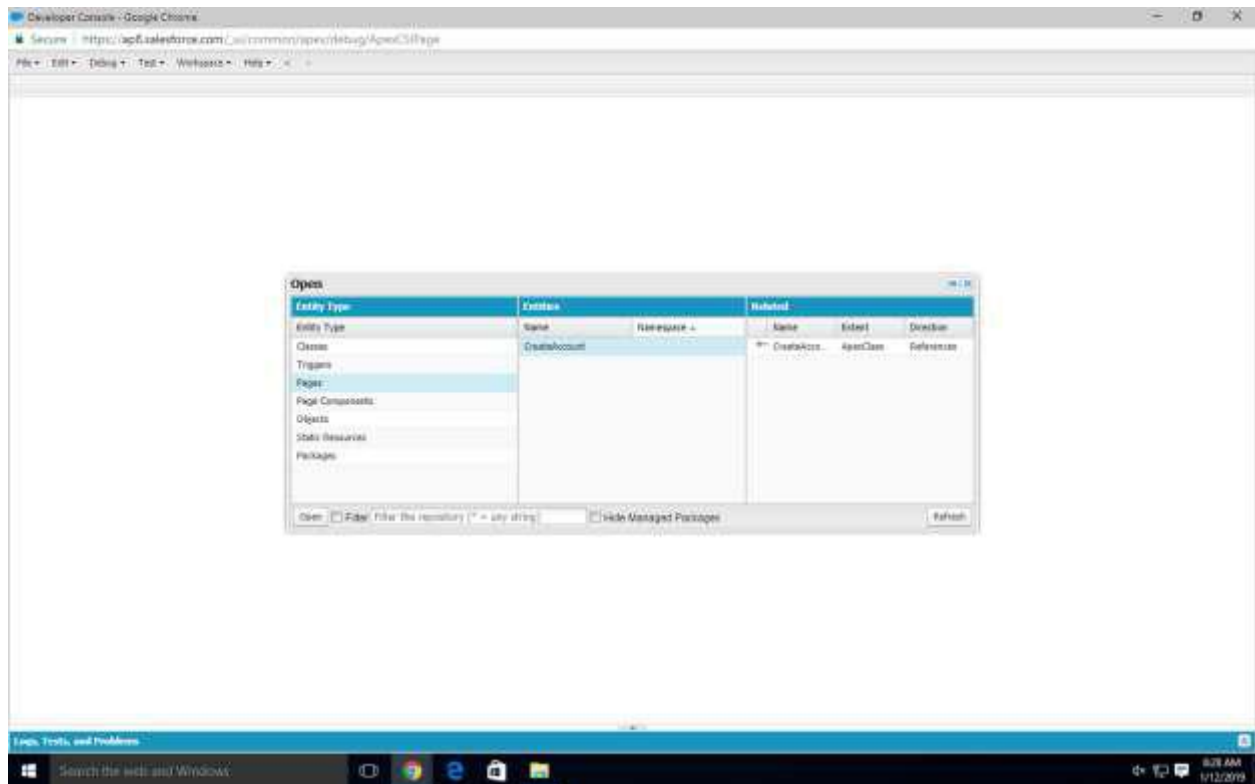
Step No 6:

Select tabs section (for creating GUI)



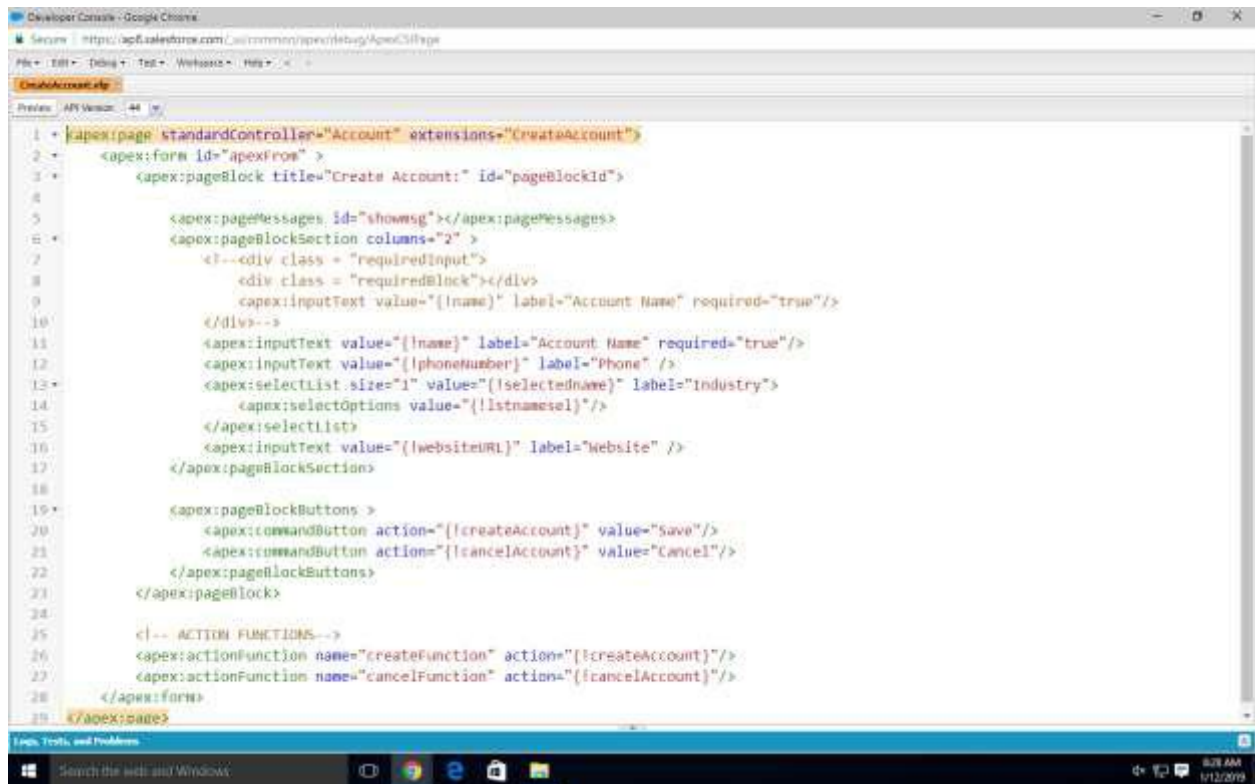


Step No 7: Open a pages



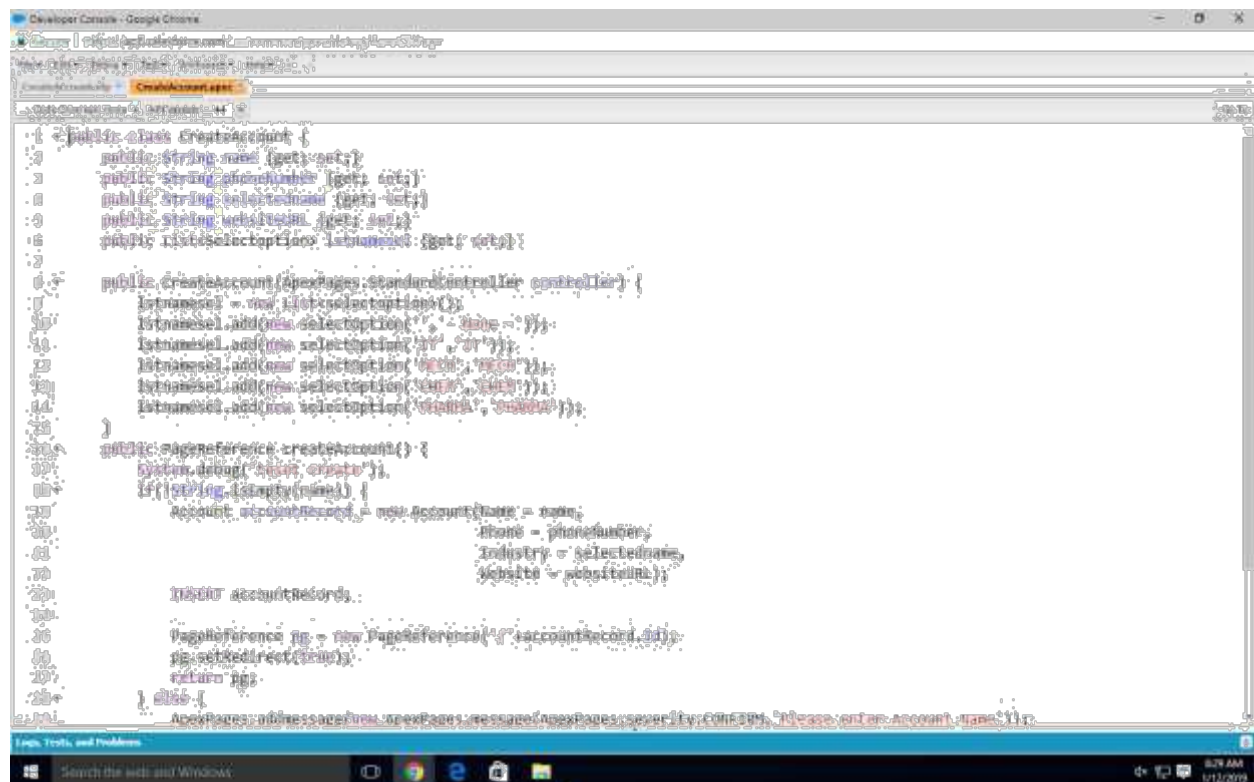
Step No 8:

Program Windows



The screenshot shows a web browser window with the URL `https://ap6.salesforce.com/.../apex/debug/Apex33Page`. The browser's developer console is open, displaying the source code of an Apex page. The code defines a page with a standard controller 'Account' and an extension 'CreateAccount'. It includes a form with fields for Account Name, Phone, Industry, and Website, along with 'Save' and 'Cancel' buttons. The form is styled with a 'requiredInput' class. The code also includes action functions for 'createFunction' and 'cancelFunction'.

```
1 <apex:page standardController="Account" extensions="CreateAccount">
2 <apex:form id="apexForm">
3 <apex:pageBlock title="Create Account:" id="pageBlockId">
4
5 <apex:pageMessages id="showmsg"></apex:pageMessages>
6 <apex:pageBlockSection columns="2">
7 <!--<div class = "requiredInput">
8 <div class = "requiredBlock"></div>
9 <apex:inputText value="{!name}" label="Account Name" required="true"/>
10 </div>-->
11 <apex:inputText value="{!name}" label="Account Name" required="true"/>
12 <apex:inputText value="{!phoneNumber}" label="Phone" />
13 <apex:selectList size="1" value="{!selectedname}" label="Industry">
14 <apex:selectOptions value="{!listnamesel}" />
15 </apex:selectList>
16 <apex:inputText value="{!websiteURL}" label="Website" />
17 </apex:pageBlockSection>
18
19 <apex:pageBlockButtons>
20 <apex:commandButton action="{!createAccount}" value="Save"/>
21 <apex:commandButton action="{!cancelAccount}" value="Cancel"/>
22 </apex:pageBlockButtons>
23 </apex:pageBlock>
24
25 <!-- ACTION FUNCTIONS-->
26 <apex:actionFunction name="createFunction" action="{!createAccount}" />
27 <apex:actionFunction name="cancelFunction" action="{!cancelAccount}" />
28 </apex:form>
29 </apex:page>
```



Program Screenshots

The screenshot displays the Salesforce Developer Console interface. At the top, the browser address bar shows the URL: `https://developer.salesforce.com/developertools/ide`. The Salesforce logo is in the top left, and a search bar is in the top right. Below the header, a navigation bar includes links for Home, Chatter, Libraries, Content, Subscribers, and Quick Account. The left sidebar, titled 'Recent Items', lists several items: Personal Library, Library, Test Account, Test, Home, Help, and Recycle Bin. The main workspace area is titled 'Create Account:' and contains a form with the following fields: Account Name, Phone, Website, and Industry (set to 'None'). There are 'Save' and 'Cancel' buttons at the bottom of the form. The footer of the page displays copyright information: Copyright © 2000-2019 Salesforce.com, Inc. All rights reserved. | Privacy Statement | Security Statement | Terms of Use | Salesforce.com | Get the Salesforce mobile app.

Output

The screenshot shows a Salesforce user interface for an account named 'prasad'. The browser address bar shows a secure URL: <https://ap8.salesforce.com/10100000002C806>. The navigation bar includes links for Home, Chatter, Libraries, Content, Subscriptions, and Quick Account. On the left, a sidebar contains a 'Create New...' dropdown, a 'Recent Items' list with links to 'prasad', 'Prasad Lahari', 'Prasad', 'Test Account', 'Test', 'Prasad', and 'Business Process Core of Oracle', and a 'Recycle Bin' link. The main content area features a profile card for 'prasad' with a 'New Feed' button and a 'Click to add topics' link. Below this is a 'Post' section with a text input field and a 'Share' button. A 'Follow' button is also present. The 'Account Detail' section is visible, showing fields for Account Owner, Account Name, Parent Account, Account Number, Account Site, Type, Industry, Annual Revenue, Billing Address, Customer Priority, SLA Expiration Date, Number of Locations, Active, Created By, Description, Custom Link, Rating, Phone, Fax, Website, Ticket Symbol, Ownership, Employee, SIC Code, Shipping Address, SLA, SLA Start/End Date, Opened Opportunity, and Last Modified By. The 'Account Detail' section is currently expanded, showing the following information:

Field	Value
Account Owner	Prasad Lahari (Chatter)
Account Name	prasad (View Hierarchy)
Parent Account	
Account Number	
Account Site	
Type	
Industry	IT
Annual Revenue	
Billing Address	
Customer Priority	
SLA Expiration Date	
Number of Locations	
Active	
Created By	Prasad Lahari, 1/11/2019 3:38 AM
Description	
Custom Link	prasad
Rating	
Phone	1234567890
Fax	
Website	http://prasad.com
Ticket Symbol	
Ownership	
Employee	
SIC Code	
Shipping Address	
SLA	
SLA Start/End Date	
Opened Opportunity	
Last Modified By	Prasad Lahari, 1/11/2019 3:38 AM